下载APP

春节特别放送3 | 揭晓项目作业的答案

2021-02-16 尹会生

Python自动化办公实战课 进入课程 >

讲述：尹会生
时长 11:20 大小 10.40M

前两节课，我给你策划了一个项目作业，也给你提供了解题思路和方法，那么今天，就是揭晓答案的时候了。不管你有没有完成作业，或者完成了多少，我都建议你在学习今天这节课的时候，一边看我的讲解，一边把你的答案与我讲解的答案进行对照。

不过我也要提醒你，因为现在你才学到了第三节课，所以答案对你来说并不是最重要的。重要的是你对前三节课的内容掌握了多少，以及这个项目作业的解题思路和方法。

话不多说，我们现在就开始吧。

第一题

我们先看第一道题中关于订单统计的处理思路。

首先我们要明确要解决的问题。

我们希望对多个 Excel 进行统计，并取得每个 Excel 文件里的 sheet。然后再从每一个 sheet 当中取出水果的名称和它的每一次销售金额，最终要将多个 Excel 里的所有的数据进行合并。而合并之后就是我们想要的全年水果销售金额统计结果。

所以解决这个问题的关键点有两个。

第一个关键知识点是遍历。这个知识点在我们的第 1 讲、第 2 讲中都有提到。通过遍历的方式，我们能依次取到每一个文件里的每一个 sheet，甚至能从每个 sheet 取出每一行的内容。

** 第二个关键的知识点就是**多个元素的累加操作。**** 主要就是把相同名称的水果销售金额进行计算。不过这里涉及到的计算很简单，就是累加操作，通过循环就可以实现多个元素的依次累加了。由于累加是针对每一行数据的，所以我们要把累加操作写在遍历每一行的循环当中。

在搞清楚了两个关键知识点之后，**接下来我来带你去分析一下我实现全年销售统计的代码。**

复制代码

```
import xlrd
from pathlib import Path, PurePath
from collections import defaultdict

# 订单路径
download_path = '/Users/edz/Desktop/效率专栏/新年特辑/订单'

# 取得该目录下所有的xlsx格式文件
p = Path(download_path)
files = [x for x in p.iterdir() if PurePath(x).match('*.xlsx')]

# 定义字典用于结果统计
total = defaultdict(int)

# 中文做字典的key会有问题,做两个简单的翻译函数
tran_dict =  {
    "dragon_fruit":"火龙果",
    "coconut":"椰子",
    "watermelon":"西瓜"
}
```

```
# 中文翻译成英文
def dict_trans_chi2eng(value):
    return [k for k,v in tran_dict.items() if v == value]

# 英文翻译成中文
def dict_name_eng2chi(key):
    return tran_dict[key]

# 遍历文件
for file in files:
    sheet = xlrd.open_workbook(file)

    # 遍历表格
    for table in  sheet.sheets():
        # 从第二行遍历内容
        for line in range(1,table.nrows):
            fruit = table.row_values(rowx=line, start_colx=0, end_colx=None)
            # 第一列水果名称，倒数第二列销售金额
            # print(f'{fruit[0]},{fruit[-2]}')
            # 火龙果,15.0
            # 椰子,16.0
            # 西瓜,10.5

            # 统计每种水果的销售额
            fruit_name = dict_trans_chi2eng(fruit[0])[0]
            total[fruit_name] = total[fruit_name] + fruit[-2]

for fruit_name in total:
    print(f'水果: {dict_name_eng2chi(fruit_name)} , 总金额: {total[fruit_name]}'

    # 水果: 火龙果 , 总金额: 135.0
    # 水果: 椰子 , 总金额: 144.0
    # 水果: 西瓜 , 总金额: 94.5
```

我们来看代码的前 3 行。第 1 行是读取 Excel 文件的库，第 2 行是用来去处理路径的库，这两个库在我们课程的第一讲都已经为你重点讲解过。**读取Excel 使用的是 xlrd 库, 处理路径需要使用 pathlib**。之后的课程你也会多次看到它们，所以没有记住的话一定要再去复习。

我重点讲解一下第 3 行出现的库。第 3 行的库是一个叫做 defaultdict 的特殊字典。它特殊在哪里呢？**和字典相比**，defaultdict 可以在**字典**初始化的时候**得到一个默认的**值。在我的代码中，就使用 defaultdict 记录了水果数量，默认把水果数量设置为 0，那就不用像默认字典一样，设置了字典之后还要手动把字典的值赋值为 0。

这三行都是程序引入的库，**接下来我们来看具体的代码逻辑。**

首先我们需要**获得 Excel 文件所在的路径**。第 9、10 行代码是用来读取目录下所有的 Excel 文件的，获取路径使用的是第 2 讲的 pathlib 库，我在第 2 讲也为你详细剖析过它的逻辑，这里就不再赘述了。

获得路径之后，接着就要去**处理每一个文件**了。从**第 30 行**开始就是我们**程序的核心逻辑**，像咱们的第二讲一样，我在这里使用了三个 for 循环。

第 1 个循环用来遍历所有的 Excel 文件。

第 2 个循环用来遍历所有的 sheet。

第 3 个循环用来读取文件当中的，第 2 行到最后一行。

通过这些循环我就可以按行进行处理。那再接着需要处理的就是**如何进行水果销售金额的累加。**

如果你仔细观察代码就会发现，我并没有马上去进行累加，而是在这里做了一个**把中文转换成英文的额外处理**，这个额外的处理使用了第三讲学过的**自定义函数功能**。

你可能会问，为什么要进行中文转换英文呢？因为在 Python 程序当中，字典这个数据类型对中文支持并不友好，使用中文作为字典的 key 进行操作时会报错。

所以为了避免字典在处理中文的时候出现报错，我在这里做了一个**中文英文映射和转换**的一个小功能。这个功能被我写在文件当中的第 15 到第 27 行，你会看到这里有一个**被定义为 translate_dict 的一个映射关系的字典**。

在这个字典当中，Key 就是英文单词，value 就是中文汉字。如果在你编写的程序当中需要支持更多的水果种类，那你就可以按照我的格式在这个字典中继续去扩展中英文的映射关系。

那中英文转换又是怎么实现的呢？我在字典之后又自定义了两个函数，分别被我命名为 Chinese to English 和 English to Chinese。

在有了中英文转换功能，那我在进行水果的累加之前，“Chinese to English”函数就可以把字典的“key”从中文的水果名称转换成英文。转换成英文，就意味着英文的水果名称可以作为统计总金额的字典 total 的 key。那么：

使用 total[fruit_name]方法就可以取得已经统计的水果金额。

使用 fruit[-2]取得通过当前得到的水果金额。

使用 total[fruit_name] = total[fruit_name] + fruit[-2] 方式，就可以实现全年的水果销售总金额统计功能。

最后，在第 49 行**把销售金额统计结果进行输出**。由于水果名称在处理过程被处理成英文，为了让你查看的结果更加友好，我还需要再做两件事情。

1. 第一个是我在输出前把英文的水果名称使用“English to Chinese”函数转换成了中文。
2. 第二个是使用了字符串连接功能，将输出的内容进行对齐并增加必要的提示信息，这样你就可以看到每种水果今年的销售总额了。

以上就是我在实现全年水果销售统计的问题上，是如何编码来实现的。

第二题

接下来我们再看第二个问题。第二个问题要求你按月进行水果销售额 Top3 的统计信息，并且取出当月销售数量前三名的水果名称。

针对这道题，在进行编码之前我们要思考一下，按月统计数据和全年统计数据有什么区别？怎么才能取出销售数量的前三名呢？

咱们首先来解决按月统计数据和全年统计数据有什么区别这个问题。

回顾一下我们在解决全年统计问题的过程。我们先取得了每次销售数据。之后又使用了 total 字典类型，统计了所有行的结果，通过循环, 将所有行的结果汇总成日数据和月数据, 又继续通过循环将月数据统计为全年销售总额。

如果需要统计水果销售额 Top3, 我们也需要先将数据统计为月销售数据, 之后不能将月数据合并到一起, 而是需要对每个月的数据进行排序, 取出销售最多的前三个水果销售额, 所以统计月销售数据以前的程序逻辑是相同的, 区别是统计月销售数据以后, 一个是进行合并一个是进行排序操作。

那第二个问题，怎样去取前三名呢？我们要想取出前几名，前提是要先进行数据的排序工作。因为排序之后才能从大到小取出销售量最高的前三名。为了实现排序这一功能，我们必须要掌握 Python 的**排序和截取前三个元素的程序编写方法**。

所以接下来我们要学习的就是具体编码。由于和统计总金额的代码功能非常相似，我们就直接复用统计总金额的获取文件路径，统计月销售额的代码。由于每一个文件代表一个月份的销售数据，所以我在第 48 行用输出文件名的方式告诉你当前在统计的文件是哪一个月份的。

通过上面的逻辑方法，我们实现了按月统计销售额和输出当前月份的功能。接下来就可以进行后续的排序，并且取出字典的 value 排前 3 的数据了。你可以通过比较传统的 sort 函数进行排序，并采用复循环三次取出前 3 水果名称和销售数量。

不过我觉得采用这样的方式，代码其实不够简洁。所以我们还有另外一种简洁的方法。

在程序中，我为了避免自己需要编程实现代码逻辑，就直接引入了 collections 标准库的 Counter 包实现排序并提取前 n 个数的功能。Counter 包支持对列表和字典进行排序、数量统计、取前 n 个数的功能，非常强大。

那这里就用它来实现基于字典的 value 从大到小进行排序，排序之后取前三个元素的功能来实现，从而取得水果销售额 Top3。

在代码的第 53 行，就是我使用了 Counter 包实现对 total 变量进行排序的具体实现代码。

在排序之后，接下来的问题就是如何取出前三个元素了。由于提取前 N 个元素的问题，Counter 包已经内置了该功能，所以在第 58 行，我可以再通过 most_common() 函数，为函数增加数字“3”作为参数，从而实现提取销量 Top3 的功能。

复制代码

```
# 使用Counter函数排序和统计数量
sorted_total = Counter(total)

# 清空本月统计数据
total = defaultdict(int)

# 通过most_commnon函数排序取出Top3
```

```
print(sorted_total.most_common(3))
```

在这个程序当中，我们也有一个要注意的事项。因为每个月份统计好的数据都会放到 total 变量当中，因此在进行下一个月统计的时候，total 变量会把本月统计结果累加上去，这样就会导致下一个月统计的数量不准确。

所以我们在统计完每一个月的统计数据之后，需要把 total 变量重新初始化。为了能够让你注意这件事情，我特意把它写在了第 56 行并详细进行了标注。这也是在进行数据统计时极容易犯错误的地方，希望你能注意它的用法。

以上就是我实现所有水果的金额统计和按月统计的实现方法，希望你能通过这两个代码理解如何去处理大批量重复的 Excel 工作。

好了，我们的春节策划到这节课就结束了，你的答案和我的一样吗？希望你能借助这个项目，查漏补缺，多多巩固前面的内容。下节课，我们继续来自动化办公的方法，我们下次见。

提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 春节特别放送2 | 用自顶至底的思路解决数据统计问题

下一篇 04 | 函数与字典:如何实现多次替换

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。